Statistical Analysis of Minimum Classification Error Learning for Gaussian and Hidden Markov Model Classifiers

Mohamed Afify, Xinwei Li, and Hui Jiang, *Member, IEEE*

***Abstract*—Minimum classification error learning realized via generalized probabilistic descent, usually referred to as (MCE/GPD), is a very popular and powerful framework for building classifiers. This paper first presents a theoretical analysis of MCE/GPD. The focus is on a simple classification problem for estimating the means of two Gaussian classes. For this simple algorithm, we derive difference equations for the class means and decision threshold during learning, and develop closed form expressions for the evolution of both the smoothed and true error. In addition, we show that the decision threshold converges to its optimal value, and provide an estimate of the number of iterations needed to approach convergence. After convergence the class means drift towards increasing their distance to infinity without contributing to the decrease of the classification error. This behavior, referred to as mean drift, is then related to the increase of the variance of the classifier. The theoretical results perfectly agree with simulations carried out for a two-class Gaussian classification problem. In addition to the obtained theoretical results we experimentally verify, in speech recognition experiments, that MCE/GPD learning of Gaussian mixture hidden Markov models qualitatively follows the pattern suggested by the theoretical analysis. We also discuss links between MCE/GPD learning and both batch gradient descent and extended Baum–Welch re-estimation. The latter two approaches are known to be popular in large scale implementations of discriminative training. Hence, the proposed analysis can be used, at least as a rough guideline, for better understanding of the properties of discriminative training algorithms for speech recognition.**

***Index Terms*—Convergence analysis, discriminative learning, generalized probabilistic descent, hidden Markov models, minimum classification error, speech recognition.**

I. Introduction

A general paradigm for the design of classifiers, based on the idea of minimizing the classification error, was proposed in [12] and [16]. In this framework, a smoothed estimate of the classification error is first formulated and is then minimized, with respect to the parameters of interest, using gradient descent. Thus, this approach is often referred to as minimum classification error/generalized probabilistic descent (MCE/GPD). Since its introduction, this learning approach has found great success in many practical, and often large-scale, classification problems. A large part of these applications focused on speech recognition and other natural language processing tasks. Interesting reviews of the MCE/GPD framework, that cover both theory and applications, can be found in [5] and [15].

Due to the success of MCE/GPD in many practical classifier design problems there were attempts to theoretically analyze its performance. We mention here the works in [4], [17], [28]. The relationship between some of these works and the current work will be highlighted in the paper. In addition to the above works, the main theoretical justification for the use of the MCE/GPD paradigm is the generalized probabilistic descent (GPD) theorem (refer to [12] for example). This theorem states that the update equations lead to a decrease of the expected value of the smoothed error function and converge to one of its local minima under some regularity assumptions. The goal of this paper is a more detailed study of the evolution of the classifier parameters and the objective function during learning.

In order to address these theoretical questions in more detail, we first focus on a simple learning scenario. The MCE/GPD algorithm is used to learn the means of a Gaussian classifier for a two-class problem. This setting leads to a relatively simple learning algorithm that is amenable to detailed theoretical study. Our main theoretical contributions are as follows.

- Detailed difference equations for the evolution of the class means and decision threshold during learning. These equations are used to prove that the threshold converges to its optimal value for a sufficiently small constant step size, and to obtain an estimate of the number of iterations needed to approach the optimal value. This convergence result is contrasted with GPD convergence [12] in the paper.
- Expressions for the smoothed and true error. Using these expressions, it is shown that the true error converges to its optimal value and that additional iterations after convergence only reduce the smoothed error and lead to increase the distance between inter-class means without reducing the true error. This is referred to as mean drift in the paper.
- An expression for the classifier variance during learning. This expression is used to establish that further iterations after threshold convergence will increase the classifier variance due to the mean drift. This is clearly a negative effect that needs to be avoided in practice.

The proposed statistical analysis of the algorithm is based on a framework for the analysis of adaptive algorithms with nonlinearities which was initially proposed in [3] and since then has

Manuscript received October 29, 2006; revised May 27, 2007. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Bill Byrne.

M. Afify is with the IBM T. J. Watson Research Center, Yorktown Heights, NY 10598 USA.

X. Li is with Nuance, Inc., Burlington, MA 01803 USA (e-mail: xwli@cse.yorku.ca).

H. Jiang is with York University, Toronto, ON M3J 1P3, Canada.

Digital Object Identifier 10.1109/TASL.2007.903304

been used in similar contexts. However, the technical details and results in this paper are related to the MCE/GPD approach and differ from other works in the adaptive signal processing literature.

In addition to the theoretical interest of the analysis, it has some links to discriminative learning of Gaussian mixture hidden Markov models that are very popular in speech recognition. First, the analyzed algorithm can be readily identified as a special case of MCE/GPD learning of the means of Gaussian mixture hidden Markov models (HMMs) [13]. It is experimentally verified in the paper, using a speech-recognition setting, that MCE/GPD learning of Gaussian mixture HMMs qualitatively follows the learning pattern suggested by the theoretical analysis. Second, it is shown in the paper that the proposed analysis can be extended to batch gradient optimization of the MCE objective function. This batch scheme is popular in large scale implementations for speech recognition [18], and is also closely related to the very popular extended Baum–Welch re-estimation scheme [29] as described in [26]. Thus, the proposed analysis, can be used, at least as a rough guideline, for better understanding of the properties of large scale discriminative training for speech recognition.

The rest of the paper is organized as follows. The Gaussian classification problem and the associated discriminative learning algorithm are formulated in Section II. Section III contains the analysis results of the algorithm of Section II. Experimental results are given in Section IV. We first provide simulation for a two-class Gaussian classifier which perfectly agrees with the theoretical findings developed in the paper. We then report on discriminative learning of Gaussian mixture hidden Markov models for E-set speech recognition, and show that the learning follows the same pattern predicted in the theoretical analysis. Relationships to other learning paradigms are discussed in Section V. Finally, the obtained results are summarized and possible future directions are discussed in Section VI. A summarized version of this paper appeared in [2].

II. Algorithm Formulation

This section presents a simple classification problem for which the discriminative training algorithm of [12] will be formulated and analyzed. Assume we have two classes C_j, where $j \in \{0,1\}$. For class C_j, the probability density function (pdf) of the observations is Gaussian given by $\mathcal{N}\left(\theta_j, \sigma^2\right)$, where θ_j is the mean for class C_j, and σ^2 is a common variance. It is worth noting that these are the true densities of the two classes. The choice of this simple setting with one-dimensional observations and Gaussian distributions facilitates the analysis of the resulting discriminative training algorithm. The use of a common variance leads to a linear misclassification function which allows the computation of certain expected values, as will be discussed in the paper, without making further approximations. In addition, the solution of the above classification problem is known [6]. This allows comparing the behavior of the learning algorithm to the known solution. Assuming, without loss of generality, that the two classes are equiprobable and $\theta_1 > \theta_0$. The optimal classifier reduces to comparing an observation x to a threshold $t = (\theta_1 + \theta_0)/2$, and deciding C_1 if $x > t$, and C_0 otherwise. The given value of the threshold is optimal, i.e., leads to minimum classification error, if the observations are Gaussian. Other threshold values should be used for different class distributions. Hence, the notion of "model correctness" in this simple setting reduces to the choice of the threshold value. In the Gaussian case the minimum (Bayes) error can be easily calculated as [6]

$$e_B = 0.5\left[\Phi\left(\frac{t-\theta_1}{\sigma}\right) + \Phi\left(\frac{\theta_0 - t}{\sigma}\right)\right] \tag{1}$$

where $\Phi()$ is the standard Gaussian cumulative distribution function.

We will now apply the discriminative training paradigm of [12] to learn the class means in the above classification problem. This will lead to a discriminative learning algorithm that will be the focus of the analysis in this paper. The goal of the considered algorithm is to sequentially estimate the means of the classes to minimize a smoothed classification error measure. This objective can be achieved by following the three steps in [12]. The procedure starts by defining a misclassification function for class C_j. In our case, it is straightforward to write this misclassification function as

$$\begin{aligned} d_j(x, \mu_j, \mu_{1-j}) &= \log p_{1-j}(x) - \log p_j(x) \\ &= \frac{(\mu_{1-j} - \mu_j)x}{\sigma^2} + \frac{\left(\mu_j^2 - \mu_{1-j}^2\right)}{2\sigma^2} \end{aligned} \tag{2}$$

where p_j is the pdf of class j, μ_j and μ_{1-j} represent the mean variables in the misclassification function to differentiate them from the true class means θ_j and θ_{1-j}. It can be also seen that $d_{1-j}(x, \mu_j, \mu_{1-j}) = -d_j(x, \mu_j, \mu_{1-j})$. A smooth estimate of the error for observation x is then obtained by using a sigmoid nonlinearity on the misclassification function. This step can be formulated as

$$e(x, \mu) = \Phi(d_j(x, \mu)) \quad x \in C_j \tag{3}$$

where we have used μ as shorthand to indicate means of both classes, this abbreviation will be used in the rest of the paper. In addition, we have replaced the usual sigmoid by the standard Gaussian cumulative distribution function (CDF). Both functions have very similar behavior and can be used for smoothing purposes [1], [4]. However, the latter allows the calculation of some expected values in closed form during the analysis. We note that the true error can be obtained by replacing the Gaussian CDF by a unit step function. Finally, the expected value of the smoothed error in (3) is minimized using the generalized probabilistic descent (GPD) algorithm. For our purpose, the GPD recursion can be written as

$$\mu_j(n+1) = \mu_j(n) - \epsilon \frac{\partial e(x(n+1), \mu)}{\partial \mu_j} \Big|_{\mu = \mu(n)} \tag{4}$$

where $\mu_j(n+1)$ and $x(n+1)$ are used to indicate the mean of C_j and the observation at iteration $n+1$, and ϵ is the learning rate. By using the definition of the error in (3), calculating the

derivatives, and after some simplification, we arrive at the update equations for the means, shown in (5) at the bottom of the page, where $\phi(x) = \left(1/\sqrt{2\pi}\right)\exp(-x^2/2)$. The mean update in (5) together with the associated expected values of the smoothed and true error are the focus of the analysis in Section III.

III. Algorithm Analysis

In this section, we perform statistical analysis of the algorithm given in Section II. In particular, we focus on deriving difference equations for $E[\mu_j(n+1)]$ and the associated decision threshold. These difference equations are helpful in studying the transient behavior of the learning. Hence, we refer to this part as transient analysis. In addition, we calculate expressions for the expected values of the smoothed and true error. This helps in studying the error evolution during learning and is referred to as error analysis. Following transient and error analysis, we develop an expression for the evolution of the classifier variance during learning. We refer to this as variance analysis. This section is structured as follows. Transient analysis is presented in Section III-A, error analysis is carried out in Section III-B, and variance analysis can be found in Section III-C. Finally, the assumptions used in the analysis and possible generalizations are discussed in Section III-D.

A. Transient Analysis

This subsection derives difference equations for $E[\mu_j(n+1)]$ and the associated decision threshold followed by studying the convergence behavior of the decision threshold. We start by the difference equation for class means. This is done by first evaluating $E[\mu_j(n+1)|\mu(n)]$ and then integrating out the conditioning by using an approximation proposed in [3]. To this end, we write

$$\begin{aligned} &E[\mu_j(n+1)|\mu(n)] \\ &\quad = P_j E[\mu_j(n+1)|\mu(n), x(n+1) \in C_j] \\ &\qquad + P_{1-j} E[\mu_j(n+1)|\mu(n), x(n+1) \in C_{1-j}] \end{aligned} \tag{6}$$

where P_j and P_{1-j} are the *a priori* probabilities of classes C_j and C_{1-j}, respectively. We will evaluate both expectations on the right-hand side of (6). It is shown in Appendix A that

$$\begin{aligned} &E[\mu_j(n+1)|\mu(n), x(n+1) \in C_j] \\ &\quad = \mu_j(n) + \frac{\epsilon}{\sigma\sqrt{\Delta^{(j)}\mu(n)^2 + \sigma^2}} \\ &\qquad \times \left[\delta^{(j)}\mu_j(n)\phi\left(\frac{d_j(\theta_j, \mu(n))\sigma}{\sqrt{\Delta^{(j)}\mu(n)^2 + \sigma^2}}\right)\right. \\ &\qquad\quad \left. + \frac{\Delta^{(j)}\mu(n)\sigma}{\sqrt{\Delta^{(j)}\mu(n)^2 + \sigma^2}}\psi\left(\frac{d_j(\theta_j, \mu(n))\sigma}{\sqrt{\Delta^{(j)}\mu(n)^2 + \sigma^2}}\right)\right] \end{aligned} \tag{7}$$

where $\delta^{(j)}\mu_j(n) \equiv \theta_j - \mu_j(n)$, $\Delta^{(j)}\mu(n) \equiv \mu_{1-j}(n) - \mu_j(n)$, $\psi(x) = -x\phi(x)$, σ is the common variance, and $d_j(\theta_j, \mu(n))$ is calculated as in (2). It can be similarly shown that

$$\begin{aligned} &E[\mu_j(n+1)|\mu(n), x(n+1) \in C_{1-j}] \\ &\quad = \mu_j(n) - \frac{\epsilon}{\sigma\sqrt{\Delta^{(j)}\mu(n)^2 + \sigma^2}} \\ &\qquad \times \left[\delta^{(1-j)}\mu_j(n)\phi\left(\frac{d_j(\theta_{1-j}, \mu(n))\sigma}{\sqrt{\Delta^{(j)}\mu(n)^2 + \sigma^2}}\right)\right. \\ &\qquad\quad \left. + \frac{\Delta^{(j)}\mu(n)\sigma}{\sqrt{\Delta^{(j)}\mu(n)^2 + \sigma^2}}\psi\left(\frac{d_j(\theta_{1-j}, \mu(n))\sigma}{\sqrt{\Delta^{(j)}\mu(n)^2 + \sigma^2}}\right)\right] \end{aligned} \tag{8}$$

where, in addition to the above definitions, we have $\delta^{(1-j)}\mu_j(n) \equiv \theta_{1-j} - \mu_j(n)$. We will now make the assumption $P_j = P_{1-j} = 0.5$. It is possible to choose any values for the class prior probabilities. However, in order that the analysis follows the true behavior of the algorithm, the presentation of the training examples should follow the chosen prior probabilities. Using this assumption and substituting into (6) leads to

$$\begin{aligned} &E[\mu_j(n+1)|\mu(n)] \\ &\quad = \mu_j(n) + \frac{0.5\epsilon}{\sigma\sqrt{\Delta^{(j)}\mu(n)^2 + \sigma^2}} \\ &\qquad \times \left[\delta^{(j)}\mu_j(n)\phi\left(\frac{d_j(\theta_j, \mu(n))\sigma}{\sqrt{\Delta^{(j)}\mu(n)^2 + \sigma^2}}\right)\right. \\ &\qquad\quad -\delta^{(1-j)}\mu_j(n)\phi\left(\frac{d_j(\theta_{1-j}, \mu(n))\sigma}{\sqrt{\Delta^{(j)}\mu(n)^2 + \sigma^2}}\right) \\ &\qquad\quad + \frac{\Delta^{(j)}\mu(n)\sigma}{\sqrt{\Delta^{(j)}\mu(n)^2 + \sigma^2}}\left(\psi\left(\frac{d_j(\theta_j, \mu(n))\sigma}{\sqrt{\Delta^{(j)}\mu(n)^2 + \sigma^2}}\right)\right. \\ &\qquad\quad \left.\left. -\psi\left(\frac{d_j(\theta_{1-j}, \mu(n))\sigma}{\sqrt{\Delta^{(j)}\mu(n)^2 + \sigma^2}}\right)\right)\right]. \end{aligned} \tag{9}$$

Now to calculate $E[\mu_j(n+1)]$ from the above expectation, we need to specify $p(\mu(n))$, to integrate out the conditioning, which is not a simple task. Instead, we follow the approach of [3] and assume that $\mu(n)$ is concentrated at $E[\mu(n)]$. This is a reasonable assumption for small step size ϵ, and has been used in many works on the analysis of adaptive algorithms. Using this assumption and denoting $E[\mu_j(n+1)] = \overline{\mu_j(n+1)}$, we get the difference equation shown in (10), at the bottom of the next page, where we have used the following definitions:

$$\overline{\delta^{(j)}\mu_j(n)} = \theta_j - \overline{\mu_j(n)} \tag{11a}$$

$$\overline{\delta^{(1-j)}\mu_j(n)} = \theta_{1-j} - \overline{\mu_j(n)} \tag{11b}$$

$$\overline{\Delta^{(j)}\mu(n)} = \overline{\mu_{1-j}(n)} - \overline{\mu_j(n)}. \tag{11c}$$

$$\mu_j(n+1) = \begin{cases} \mu_j(n) + \frac{\epsilon}{\sigma^2}\phi(d_j(x(n+1), \mu(n)))(x(n+1) - \mu_j(n)), & \text{if } x(n+1) \in C_j, \\ \mu_j(n) - \frac{\epsilon}{\sigma^2}\phi(d_j(x(n+1), \mu(n)))(x(n+1) - \mu_j(n)), & \text{if } x(n+1) \in C_{1-j} \end{cases} \tag{5}$$

The difference equation in (10) describes the evolution of the mean during learning. We then define $\overline{t(n)} = (\overline{\mu_1(n)} + \overline{\mu_0(n)})/2$ as the evolution of the decision threshold during learning. Using (10), for $j = 0, 1$, together with some algebraic simplifications we arrive at the following recursion for the decision threshold:

$$\overline{t(n+1)} = \overline{t(n)} + \frac{0.25\epsilon\overline{\Delta^{(1)}\mu(n)}}{\sigma\sqrt{\overline{\Delta^{(1)}\mu(n)}^2 + \sigma^2}} \times \left[\phi\left(\frac{\overline{\Delta^{(1)}\mu(n)}(\theta_1 - \overline{t(n)})}{\sigma\sqrt{\overline{\Delta^{(1)}\mu(n)}^2 + \sigma^2}}\right) - \phi\left(\frac{\overline{\Delta^{(1)}\mu(n)}(\overline{t(n)} - \theta_0)}{\sigma\sqrt{\overline{\Delta^{(1)}\mu(n)}^2 + \sigma^2}}\right)\right] \quad (12)$$

where we have used the following relationships in the derivation, $\overline{\Delta^{(1)}\mu(n)} = -\overline{\Delta^{(0)}\mu(n)}$, and hence $\overline{\Delta^{(1)}\mu(n)}^2 = \overline{\Delta^{(0)}\mu(n)}^2$, $d_1(x, \overline{\mu(n)}) = -d_0(x, \overline{\mu(n)})$, $\phi(x)$ is an even function, $\psi(x)$ is an odd function, and $d_1(x, \overline{\mu(n)})\sigma = \overline{\Delta^{(1)}\mu(n)}(x - \overline{t(n)})/\sigma$.

Now when $n \to \infty$ and the threshold has converged, we have $\overline{t(n+1)} = \overline{t(n)} = t^*$. The steady-state threshold t^* can be calculated by equating the second term on the right-hand side of (12) to zero. After simple calculations we get, excluding the case that[1] $\overline{\Delta^{(1)}\mu(n)} = 0$, $t^* = (\theta_1 + \theta_0)/2 = t$. Hence, at steady state, the decision threshold will converge to its optimal value. Once the threshold reaches its optimal value, it will remain there. This can be readily seen from (12). Theorem 1 below formulates the result that the threshold indeed converges to its optimal value, and provides an estimate of the number of iterations needed to approach this value. The theorem is proven in Appendix D. Before stating the theorem, we give the following definitions:

$$\Delta\theta \equiv \frac{(\theta_1 - \theta_0)}{2} = \theta_1 - t = t - \theta_0 \quad (13)$$

$$\alpha(n) \equiv \frac{\overline{\Delta^{(1)}\mu(n)}}{\sqrt{\overline{\Delta^{(1)}\mu(n)}^2 + \sigma^2}}. \quad (14)$$

[1]This implies the two class means coincide.

Note that if $\overline{\mu_1(n)} > \overline{\mu_0(n)}$, we have $\alpha(n) < 0$ from the definition in (11c). Recall that we assumed that $\theta_1 > \theta_0$. Thus, the previous condition can be guaranteed using proper initialization. Also note that we always have $|\alpha(n)| \leq 1$ from (14), and that $|\alpha(n)|$ is an increasing function of n, because the learning algorithm will always increase the ratio $|\overline{\Delta^{(1)}\mu(n)}/\sigma|$ as discussed in Section III-B. These properties will be used in proving threshold convergence.

$$\Delta t(n) = \overline{t(n)} - t. \quad (15)$$

This is the distance of the threshold at iteration n from the optimal threshold. In proving Theorem 1, we assume $\Delta t(n) > 0$. The case $\Delta t(n) < 0$ can be similarly proven.

Theorem 1: Given the definitions in (13)–(15), and if $\gamma \equiv (0.5\epsilon|\alpha^3(0)|\Delta\theta/\sqrt{2\pi}\sigma^3) < 1$, the update in (12) will asymptotically converge to the optimal threshold value. Moreover, if γ is sufficiently small, the number of iterations needed to approach the optimal threshold value is given by $N \approx 1/\gamma$.

The main condition of the theorem can be guaranteed by appropriately selecting the step size and proper initialization. Also, the number of iterations needed to approach the optimal value is determined by these two parameters. It is interesting to compare this convergence result to the GPD theorem. Here, we assume only a sufficiently small constant step size and proper initialization to establish asymptotic convergence. In addition, we estimate the number of iterations needed to approach the optimal threshold. This is in contrast to GPD convergence where a decreasing step size that should satisfy certain properties is needed to prove convergence. However, the GPD result applies to more general conditions than studied here.

We note that the convergence of the decision threshold to its optimal value does not uniquely determine the class means. In fact, any values of the class means that satisfy the equation $t = (\overline{\mu_1(n)} + \overline{\mu_0(n)})/2$ are possible. This agrees with [17] and [28], where it is shown that for model-free optimization, the MCE criterion results in a distribution which leads to the same classification error as the true distribution. Indeed, any two distributions with a common variance and whose means satisfy the previous condition will lead to the same minimum error. Error analysis in Section III-B will be used to study in more detail the behavior of the class means during learning.

$$\overline{\mu_j(n+1)} = \overline{\mu_j(n)} + \frac{0.5\epsilon}{\sigma\sqrt{\overline{\Delta^{(j)}\mu(n)}^2 + \sigma^2}} \times \left[\overline{\delta^{(j)}\mu_j(n)}\phi\left(\frac{d_j(\theta_j, \overline{\mu(n)})\sigma}{\sqrt{\overline{\Delta^{(j)}\mu(n)}^2 + \sigma^2}}\right) - \overline{\delta^{(1-j)}\mu_j(n)}\phi\left(\frac{d_j(\theta_{1-j}, \overline{\mu(n)})\sigma}{\sqrt{\overline{\Delta^{(j)}\mu(n)}^2 + \sigma^2}}\right) + \frac{\overline{\Delta^{(j)}\mu(n)}\sigma}{\sqrt{\overline{\Delta^{(j)}\mu(n)}^2 + \sigma^2}}\left(\psi\left(\frac{d_j(\theta_j, \overline{\mu(n)})\sigma}{\sqrt{\overline{\Delta^{(j)}\mu(n)}^2 + \sigma^2}}\right) - \psi\left(\frac{d_j(\theta_{1-j}, \overline{\mu(n)})\sigma}{\sqrt{\overline{\Delta^{(j)}\mu(n)}^2 + \sigma^2}}\right)\right)\right] \quad (10)$$

B. Error Analysis

In this section, we derive expressions for the expectations of the smoothed error $E[e(x,\mu(n))]$, and the true error $E[e_T(x,\mu(n))]$. In addition, we discuss some of their properties which will help in studying the behavior of the learning. We start by the expectation of the smoothed error. Using (3), we write

$$E[e(x,\mu(n))|\mu(n)] = P_j E[e(x,\mu(n))|\mu(n), x\in C_j] + P_{1-j}E[e(x,\mu(n))|\mu(n), x\in C_{1-j}]. \quad (16)$$

Evaluating the first expectation on the right-hand side of (16), we have

$$\begin{aligned} &E[e(x,\mu(n))|\mu(n), x\in C_j] \\ &\quad = E[\Phi(d_j(x,\mu(n)))|\mu(n), x\in C_j] \\ &\quad = E\left[\Phi\left(\left(\frac{|\Delta^{(j)}\mu(n)|}{\sigma}\right)v_j + d_j(\theta_j,\mu(n))\right)|\mu(n), x\in C_j\right] \\ &\quad = \Phi\left(\frac{d_j(\theta_j,\mu(n))\sigma}{\sqrt{\Delta^{(j)}\mu(n)^2+\sigma^2}}\right) \end{aligned} \quad (17)$$

where the first line follows from the definition of error in (3), the second line from the definitions of (38) and (39) and in Appendix A, and the third line by noting that v_j has zero mean and unit variance, and using the normal identity $\int_{-\infty}^{\infty}\Phi(ax+b)\phi(x)dx = \Phi(b/\sqrt{1+a^2})$ [23]. It can be similarly shown that

$$E[e(x,\mu(n))|\mu(n), x\in C_{1-j}] = \Phi\left(\frac{d_{1-j}(\theta_{1-j},\mu(n))\sigma}{\sqrt{\Delta^{(j)}\mu(n)^2+\sigma^2}}\right). \quad (18)$$

Substituting (17) and (18) into (16), and assuming, as in the previous subsection, equiprobable class priors ($P_j = P_{1-j} = 0.5$), we obtain

$$E[e(x,\mu(n))|\mu(n)] = 0.5\left[\Phi\left(\frac{d_j(\theta_j,\mu(n))\sigma}{\sqrt{\Delta^{(j)}\mu(n)^2+\sigma^2}}\right) + \Phi\left(\frac{d_{1-j}(\theta_{1-j},\mu(n))\sigma}{\sqrt{\Delta^{(j)}\mu(n)^2+\sigma^2}}\right)\right]. \quad (19)$$

Further making the assumption, of the previous subsection, that $\mu(n)$ is concentrated at $E[\mu(n)]$, and denoting $E[\mu(n)] = \overline{\mu(n)}$, and $E[e(x,\overline{\mu(n)})] = \overline{e(n)}$, we arrive at

$$\overline{e(n)} = 0.5\left[\Phi\left(\frac{d_j(\theta_j,\overline{\mu(n)})\sigma}{\sqrt{\overline{\Delta^{(j)}\mu(n)}^2+\sigma^2}}\right) + \Phi\left(\frac{d_{1-j}(\theta_{1-j},\overline{\mu(n)})\sigma}{\sqrt{\overline{\Delta^{(j)}\mu(n)}^2+\sigma^2}}\right)\right]. \quad (20)$$

In the above analysis, we obtained an expression for the expected value of the smoothed error as given in (20). We are also interested in obtaining the expected value of the true error $(e_T(x))$, often referred to as the 0–1 loss. This can be easily obtained by replacing $\Phi()$ in (3) by a unit step and explicitly evaluating the resulting integral. For example, we have

$$\begin{aligned} &E[e_T(x,\mu(n))|\mu(n), x\in C_j] \\ &\quad = E[U(d_j(x,\mu(n)))|\mu(n), x\in C_j] \\ &\quad = \Phi\left(\frac{d_j(\theta_j,\mu(n))\sigma}{|\Delta^{(j)}\mu(n)|}\right) \end{aligned} \quad (21)$$

where U is the unit step function, and the second line follows by explicitly evaluating the expectation taking into account that $d_j(x,\mu(n))$ is Gaussian with mean and variance given in (38) and (39). Further proceeding in exactly the same way as the smoothed error case, we arrive at the following expression for the expected value of $e_T(x)$:

$$\overline{e_T(n)} = 0.5\left[\Phi\left(\frac{d_j(\theta_j,\overline{\mu(n)})\sigma}{|\overline{\Delta^{(j)}\mu(n)}|}\right) + \Phi\left(\frac{d_{1-j}(\theta_{1-j},\overline{\mu(n)})\sigma}{|\overline{\Delta^{(j)}\mu(n)}|}\right)\right]. \quad (22)$$

In Section II, we assumed that $\theta_1 > \theta_0$. Further assuming that $\overline{\mu_1(n)} > \overline{\mu_0(n)}$ is preserved during the learning,[2] the error expression in (22) can be simplified to

$$\overline{e_T(n)} = 0.5\left[\Phi\left(\frac{\overline{t(n)}-\theta_1}{\sigma}\right) + \Phi\left(\frac{\theta_0-\overline{t(n)}}{\sigma}\right)\right] \quad (23)$$

where $\overline{t(n)} = (\overline{\mu_1(n)} + \overline{\mu_0(n)})/2$ can be considered as the evolution of the decision threshold during learning. Comparing (23) and (1), we find that they are similar except for replacing t by $\overline{t(n)}$. It can be easily verified from (23) that the error $\overline{e_T(n)}$ monotonically decreases as $\overline{t(n)}$ approaches t. Hence, as the decision threshold evolves towards its optimal value, as discussed in the previous section, the true error will decrease until it converges to its minimum value, the Bayes error.

Next, examining (20) and (22), and focusing on the first terms on the right-hand side, we can readily see that

$$\begin{aligned} d_j(\theta_j,\overline{\mu(n)}) \le 0 &\Longrightarrow \frac{d_j(\theta_j,\overline{\mu(n)})\sigma}{\sqrt{\overline{\Delta^{(j)}\mu(n)}^2+\sigma^2}} \\ &\ge \frac{d_j(\theta_j,\overline{\mu(n)})\sigma}{|\overline{\Delta^{(j)}\mu(n)}|} \\ &\Longrightarrow \Phi\left(\frac{d_j(\theta_j,\overline{\mu(n)})\sigma}{\sqrt{\overline{\Delta^{(j)}\mu(n)}^2+\sigma^2}}\right) \\ &\ge \Phi\left(\frac{d_j(\theta_j,\overline{\mu(n)})\sigma}{|\overline{\Delta^{(j)}\mu(n)}|}\right). \end{aligned}$$

[2]This is guaranteed given appropriate initialization.

This is because $\sqrt{\overline{\Delta^{(j)}\mu(n)}^2+\sigma^2} \geq |\overline{\Delta^{(j)}\mu(n)}|$, and $\Phi()$ is an increasing function of its argument. Similarly, it can be shown that

$$d_{1-j}(\theta_{1-j},\overline{\mu(n)}) \leq 0 \Longrightarrow \Phi\left(\frac{d_{1-j}(\theta_{1-j},\overline{\mu(n)})\sigma}{\sqrt{\overline{\Delta^{(j)}\mu(n)}^2+\sigma^2}}\right) \geq \Phi\left(\frac{d_{1-j}(\theta_{1-j},\overline{\mu(n)})\sigma}{|\overline{\Delta^{(j)}\mu(n)}|}\right).$$

Combining these results, we deduce that $\overline{e(n)} \geq \overline{e_T(n)}$ if $d_j(\theta_j,\overline{\mu(n)}) \leq 0$, and $d_{1-j}(\theta_{1-j},\overline{\mu(n)}) \leq 0$, with equality when $|\overline{\Delta^{(j)}\mu(n)}|/\sigma \to \infty$. These conditions simply state that the class means are correctly classified during the learning and are expected to hold. The latter property suggests that the smoothed error will try during the learning to reach its lower bound, the true error. To achieve this, the learning algorithm will continue to increase the ratio $|\overline{\Delta^{(j)}\mu(n)}|/\sigma$. It is interesting to relate the above upper bound result to the work in [28]. In [28], it was shown that the MCE objective function is an upper bound to the Bayes error under very general conditions. The result obtained here is a special case of that in [28], but the simplified problem allows deriving exact expressions of the error, and establishing a condition for the smoothed error to approach the true error in terms of the classifier parameters. This is not possible in the general case.

The above properties together with the transient analysis of the previous section suggest the following behavior of the learning algorithm. First, the decision threshold will move in the direction of its optimal value and will converge to this optimal value. Before convergence of the threshold, the expected true error will decrease until it reaches its minimum, the Bayes error. Once the threshold converges, it will remain at its optimal value indicating that the class means will move along the straight line $t = (\overline{\mu_1(n)} + \overline{\mu_0(n)})/2$. In spite of the saturation of the true error, the smoothed error objective function will continue to decrease by increasing the ratio $|\overline{\Delta^{(j)}\mu(n)}|/\sigma$, i.e., by moving the class means apart, until it reaches infinity. Hence, the mean values will continue to drift without contributing to an actual decrease of the true error, and only decreasing the smoothed error objective function. In Section III-C, we will derive an expression for the evolution of the variance of the classifier during learning and show that the mean drift is related to the increase of the classifier variance.

C. Variance Analysis

In this section, we first derive an expression for the evolution of the variance of the decision threshold during learning. Then we show that after convergence the mean drift, discussed in the previous subsection, is related to the increase of the variance of the classifier.

To calculate the variance of the decision threshold, we first use (5) and the definition of $t(n)$ to arrive at the following update equation of the decision threshold:

$$t(n+1) = \begin{cases} t(n)+r_j(x(n+1),\mu(n)), & \text{if } x(n+1)\in C_j \\ t(n)-r_j(x(n+1),\mu(n)), & \text{if } x(n+1)\in C_{1-j} \end{cases} \quad (24)$$

where $r_j(x(n+1),\mu(n)) = (\epsilon/2\sigma^2)\phi(d_j(x(n+1),\mu(n)))\Delta^{(j)}\mu(n)$, and $\Delta^{(j)}\mu(n)$ is defined below (7). Similar to Section III-A, we can write the following expected values conditioned on the values of the class means:

$$\begin{aligned} &E[t^m(n+1)|\mu(n)] \\ &\quad = P_j E[t^m(n+1)|\mu(n), x(n+1)\in C_j] \\ &\quad + P_{1-j} E[t^m(n+1)|\mu(n), \\ &\qquad x(n+1)\in C_{1-j}]. \end{aligned} \quad (25)$$

Using the above definitions, we show in Appendix C that

$$\begin{aligned} &Var[t(n+1)|\mu(n)] \\ &\quad = -(E[t(n+1)|\mu(n)] - t(n))^2 \\ &\qquad + P_j E\left[r_j^2(x(n+1),\mu(n))|\mu(n), x(n+1)\in C_j\right] \\ &\qquad + P_{1-j} E\left[r_j^2(x(n+1),\mu(n))|\mu(n), \right. \\ &\qquad\qquad \left. x(n+1)\in C_{1-j}\right]. \end{aligned} \quad (26)$$

Further using the definition of $r_j(x(n+1),\mu(n))$ above, we evaluate the expectations in the second and third terms of (26). This is done by a technique, similar to the second expectation in Appendix A, that is outlined in Appendix C. Finally, by combining the results, setting equal class *a priori* probabilities, and taking $\mu(n) = \overline{\mu(n)}$, as in the previous subsections, we get the following expression for the threshold variance:

$$\begin{aligned} &Var[t(n+1)|\overline{\mu(n)}] \\ &\quad = -(\overline{t(n+1)} - \overline{t(n)})^2 \\ &\qquad + \frac{\epsilon^2}{8\sqrt{2\pi}\sigma}\left(\frac{\overline{\Delta^{(j)}\mu(n)}}{\sigma}\right)^2 \frac{1}{\sqrt{2\overline{\Delta^{(j)}\mu(n)}^2+\sigma^2}} \\ &\qquad \times \left[\phi\left(\frac{\sqrt{2}d_j(\theta_j,\overline{\mu(n)})\sigma}{\sqrt{2\overline{\Delta^{(j)}\mu(n)}^2+\sigma^2}}\right)\right. \\ &\qquad\qquad \left. +\phi\left(\frac{\sqrt{2}d_{1-j}(\theta_{1-j},\overline{\mu(n)})\sigma}{\sqrt{2\overline{\Delta^{(j)}\mu(n)}^2+\sigma^2}}\right)\right]. \end{aligned} \quad (27)$$

At convergence $\overline{t(n+1)} = \overline{t(n)}$, and hence the first term in (27) will vanish leaving the second term which increases with the mean drift. This agrees with the result obtained in [20] for maximum mutual information (MMI) training which is a closely related discriminative learning algorithm. In [20], it is shown that given the correct model, MMI increases the variance of the parameter estimates. Here, the simple setting allows the derivation of a more precise expression for the variance increase in terms of the classifier parameters as given in (27). This increase of the variance needs to be avoided in practice.

D. Discussion of Assumptions and Results

The results obtained in the previous three subsections are for MCE/GPD learning of the means of two Gaussian classes with common variance under the following assumptions:

- the two classes are equiprobable;
- the step size is sufficiently small so that the mean is concentrated at its expected value.

In this section, we will discuss the above two assumptions and also possible extensions to more general estimation problems.

The first assumption does not cause a loss of generality. The analysis can be done for any values of the prior probabilities. The final conclusions regarding threshold convergence and variance will remain valid. Of course, the optimal threshold value will change with the prior probabilities. It should be noted, however, that for the algorithm to follow the theoretical analysis, the presentation of the training examples must follow the assumed prior distribution. The second assumption basically means that

$$E[\mu(n+1)] = \int E[\mu(n+1)|\mu(n)]p(\mu(n))d\mu(n) \approx E[\mu(n+1)|\overline{\mu(n)}] \tag{28}$$

where $\overline{\mu(n)}$ is the mean of $p(\mu(n))$. That is the probability mass is concentrated at the mean. One way to ensure this is to use a sufficiently small step size. In our simulation, we found that the algorithm follows the theoretical analysis for reasonable values of the learning rate. In addition, convergence of the threshold as implied by Theorem 1 requires a sufficiently small step size. Hence, this assumption does not cause any practical or theoretical problem.

It is also interesting to consider generalizations of the rather simple analyzed algorithm. If both means and variances are to be estimated, the analysis of the mean update formula can be extended in a rather simple way to a time varying variance. This can be achieved by first conditioning on the variance and then using an assumption that the variance is concentrated at its expected value similar to the assumption used for the mean in this paper. On the other hand the analysis of the variance update formula[3] is more difficult. For example, the update requires parameter transformation to maintain the positivity of the variance when applying the GPD algorithm [13]. The analysis of the variance update is not attempted in this paper. In Section IV, we will experimentally study the parameter evolution of Gaussian mixture hidden Markov models (HMMs), where both means and variances are updated.

Another popular alternative to the stochastic gradient algorithm considered in this paper is using a batch update rule. Application of this rule to large scale MCE learning can be found in [18], and it is also closely related to the popular extended Baum–Welch re-estimation formulas [26]. These links will be further explored in Section V. In this approach, the parameters are modified after averaging the gradient of a number of training examples. For the analyzed algorithm, the update of (5) will be replaced by

$$\mu_j(n+1) = \mu_j(n) + \frac{\epsilon}{N\sigma^2} \times \left[\sum_{x_i \in C_j} \phi(d_j(x_i, \mu(n)))(x_i - \mu_j(n)) - \sum_{x_i \in C_{1-j}} \phi(d_j(x_i, \mu(n)))(x_i - \mu_j(n)) \right] \tag{29}$$

where i is an index on the training examples, and N is their total number. We note that the time scale n stands for one pass over the whole data instead of a single training sample. For a sufficiently large number of examples, the above summations approach expected values which are similar to those on the right-hand sides of (7) and (8), respectively. Hence, the batch algorithm will follow the same difference equations derived for the stochastic algorithm, and hence will have the same transient behavior. The error analysis will be the same as it does not depend on the type of the update. In this paper we, however, choose to analyze the GPD algorithm due to its close links to the original implementations of the MCE/GPD framework.

IV. Experimental Results

This section first shows simulation results on a Gaussian classifier which perfectly agree with the theoretical developments. This is followed by experimentally studying the performance of MCE/GPD training of HMMs for an English E-set alphabet recognition task.

A. Results on Gaussian Data

Extensive simulation with Gaussian data was performed and in all experiments we found that the theoretical analysis perfectly predicts the behavior of the learning. We show here one example for illustration.

We generated 10000 samples from two Gaussian-distributed classes having means $\theta_1 = 2.0$ and $\theta_0 = 1.0$, and a common variance $\sigma^2 = 1.0$. The a priori class probabilities used in data generation were $P_1 = P_0 = 0.5$. We ran one pass through the data, corresponding to 10000 iterations, and used (10) to calculate the expected evolution of the means, and (20) and (22) to compute the expected values of the smoothed and true errors during learning. We also generated 100 sets of 10000 samples each, and used each set to learn the class means as in (5), and to calculate the empirical classification error at each iteration. We averaged the results of these sets to obtain the expected values of the means and classification error.

The sample based means and variance, corresponding to ML estimates in this case, were first calculated. The sample means perturbed by 0.5 and the sample variance were used to initialize the learning. Thus, the decision threshold is initially set at a distance 0.5 from its optimal value. The step size used in this experiment is $\epsilon = 0.01$.

The results shown in Figs. 1 and 2 indicate the close match of the theoretical results and the simulation. Indeed, for the evolution of the class means in Fig. 1 we notice the perfect agreement of the simulation results with the results predicted by (10). Also for the error evolution we notice the coincidence of the empirical classification error and the expected true error as predicted by (22), and the expected smoothed error is an upper bound of both as discussed in Section III-B. Both means will evolve until the threshold reaches its optimal value. At this point the true error will reach its minimum and both the error and decision threshold will not change. On the other hand, the objective function, the expected value of the smoothed error, decreases due to the increase of the ratio $|\overline{\Delta^{(j)}\mu(n)}|/\sigma$ to asymptotically reach the true

[3]This is not shown in this paper; we refer the reader to [13] for the general case of Gaussian mixture HMMs.

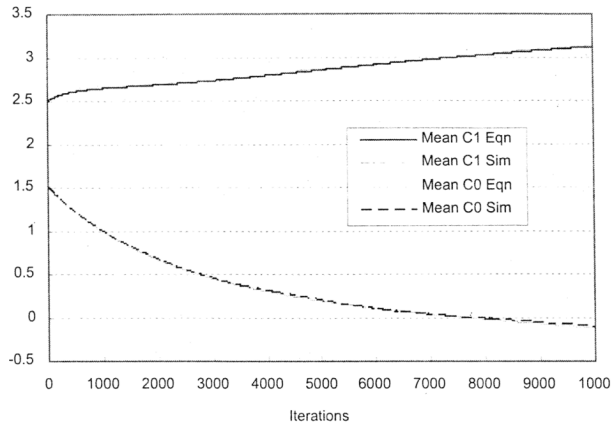

Fig. 1. Evolution of the mean of classes C_1, and C_0 during 10000 iterations of learning. Eqn denotes using (10) to calculate the mean, while Sim stands for using the learning algorithm in (5) to estimate the mean and averaging on 100 runs. The classes follow a Gaussian distribution with means $\theta_1 = 2.0$, $\theta_0 = 1.0$, and common variance $\sigma^2 = 1.0$. The learning rate is 0.01.

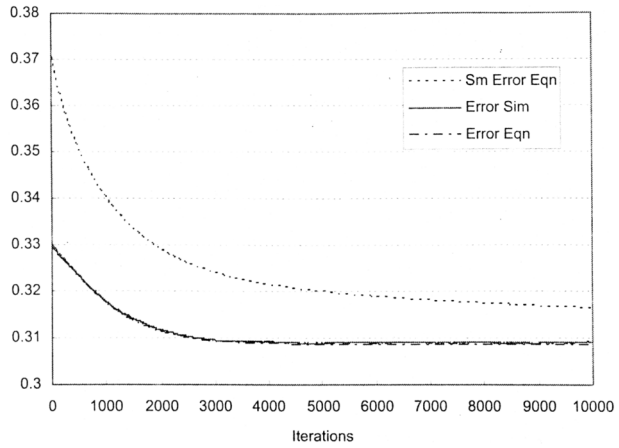

Fig. 2. Evolution of the expected value of the smoothed and true error during 10000 iterations of learning, using (20), and (22). Sim refers to averaging the empirical error in 100 runs. The classes follow a Gaussian distribution with means $\theta_1 = 2.0$, $\theta_0 = 1.0$, and common variance $\sigma^2 = 1.0$. The learning rate is 0.01.

error. This agrees with the discussion at the end of Section III-B.

Also, we have $\epsilon = 0.01$, $|\alpha(0)| = (2.5 - 1.5)/\sqrt{(2.5-1.5)^2+1} \approx 0.7$, $\Delta\theta = (2-1)/2 = 0.5$, and $\sigma = 1$. This leads to $N \approx 1/\gamma \approx 2915$ iterations from Theorem 1. Thus, the threshold will approach its optimal value in about 2915 iterations. This agrees with the simulation in Fig. 2, where the threshold, and hence the true error, appear to converge around 3000 iterations.

B. Speech Recognition Experiments

In this section, we will study the performance of MCE/GPD training of HMMs for E-set recognition. Motivated by the similarity of the Gaussian classifier update equation (5) to MCE/GPD learning of the means of Gaussian mixture HMMs [13], we may expect that the obtained theoretical results will carry over to the HMM case. Thus, we empirically study the behavior of MCE/GPD iterations for Gaussian mixture HMMs to verify the suggested learning pattern. That is, the training

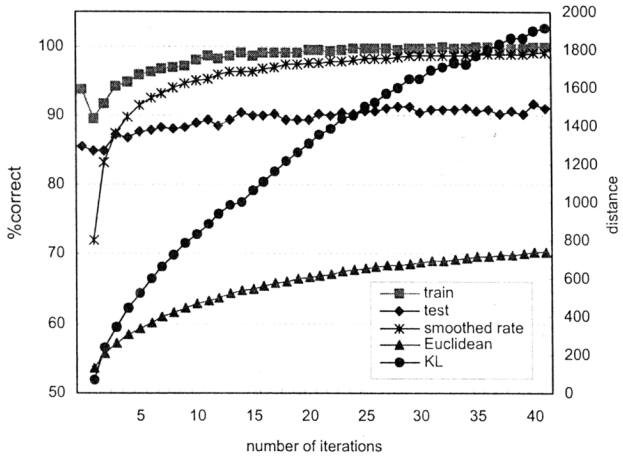

Fig. 3. MCE/GPD learning curves in HMM-based speech recognition.

set error (true error) and MCE objective function (smoothed error) will decrease, with the smoothed error being an upper bound of the true error. After the true error saturates, the MCE objective function will continue to decrease causing only the "distance" between models to increase without further reducing the classification error. Details of the experimental setup used to verify this pattern are given below.

The experiments are performed on the English E-set vocabulary of ISOLET database, consisting of $\{B, C, D, E, G, P, T, V, Z\}$. ISOLET is a database of letters of the English alphabet spoken in isolation. The database consists of 7800 spoken letters, two productions of each letter by 150 speakers. The recordings were done under quiet laboratory conditions with a noise-canceling microphone. The data were sampled at 16 kHz with 16-bit quantization. ISOLET is divided into five parts named ISOLET 1–5. In this experiment, only the first production of each letter in ISOLET 1–4 is used as training data. All data in ISOLET 5 is used as testing data. The feature vector has 39 dimensions, which include 12-d static MFCC, log-energy, delta, and acceleration coefficients.

An HMM recognizer, with 16-state, 1-mixture/state whole-word based models, is trained by HTK to be the initial models for MCE training. We chose 1-mixture/state models for efficiency purpose to be able to run many iterations on the data, and we expect that the results will generalize to multiple mixture models. This configuration achieves the best performance for the 1-mixture/state whole-word based models. The recognizer achieves an accuracy of 93.89% on the training set, and an accuracy of 85.56% on the testing data.

In the experiments the MCE/GPD discriminant function is normalized by the utterance length and feature dimension. Both means and variances will be updated for each training sample. The step size used in this experiment is $\epsilon = 3$. The weight η in the misclassification function is set to 4. The scale γ in sigmoid function is set to 2 and the shift θ is set to 0. These are parameters of MCE/GPD learning of Gaussian mixture HMMs as defined in [13]. For the E-set test, the recognition rate for the training data set is improved from 93.89% to 99.91% (only one misclassification left), while the recognition rate for the testing data set is improved from 85.56% to 91.67%. In Fig. 3, we plot the results for both the training and the testing sets as a function

of the number of iterations of the MCE training procedure. Each iteration includes updates over all the training samples.

In Fig. 3, "Train" stands for the recognition rate of the training data set, and "Test" stands for the recognition rate of the testing data set. "Smoothed Rate" stands for the smoothed recognition rate of the training data set. These three curves use the y-axis on the left side. "Euclidean" stands for the sum of Euclidean distances between each pair of models [9]. "KL" stands for the sum of Kullback–Leibler distances between each pair of models [9], [10]. These two curves use the y-axis on the right side. Note that recognition rate is used, in the figure, instead of error for convenience. Hence, for example, the "Smoothed Rate" appears as a lower bound to the "Train."

A few comments concerning the distance between the models are worth mentioning here. First, the distance is calculated as the sum of the distances between each pair of models, i.e., $D = \sum_{i,j} d_{ij}$, where the indices i, j run over all possible models. In our case, there will be $9 \times (9-1) = 72$ terms in the summation, where "9" is the number of models of the E-set. The distance between each two models, in turn, is calculated, as suggested in [9] and [10], as the sum of distances along the minimum path between each pair of states. As the states, in our case, have single Gaussians, the distance between each pair of states can be calculated in closed form either using the variance normalized Euclidean distance between the means or using the expression for the KL distance between two Gaussian distributions. These are referred to as "Euclidean", and "KL" above. If Gaussian mixtures were used for the states the calculated distance can be appropriately extended as discussed in [9] and [10].

The curve "Train" shows that the true recognition rate converges after about 25 iterations. After the convergence of the true recognition rate, the smoothed recognition rate for the training data set (curve "Smoothed Rate") continues to increase towards the true recognition rate. This agrees with what was shown in the Gaussian case, that the smoothed error objective function is an upper bound of the true error on training data. In addition, the recognition rate for testing data (curve "Test") has no apparent improvement (even drops down a little) after the true recognition rate converges. This agrees pretty well with the pattern suggested at the beginning of this section. Also, another important observation is that the distances between models continues to increase (in Euclidean and KL sense) even after the true error rate converges. This indicates the "mean drift", here better called "model drift", found in the theoretical analysis. This drift is expected to contribute to the increase of the variance of the classifier though this is not theoretically proved for the HMM case.

A few comments regarding our experimental setup are worth mentioning here. Due to our desire of keeping a practical, yet simple, speech recognition scenario, we deviated from our theoretical framework in some aspects. First, a multiple class-recognition problem was considered. It was shown in [19] how to formalize an MCE objective function using pairwise mis-classification measures, and this work can be considered as a starting point to generalize our analysis to the general multiple class problem. In this work to handle this generalization we used the pairwise averaged model distance in place of the normalized inter-class means distance $|\overline{\Delta^{(j)}\mu(n)}|/\sigma$. Second, we also chose to update both means and variances while the analysis considered only mean estimation for fixed variance.

V. Links to Other Discriminative Objective Functions and Optimization Criteria

Recently the most popular approach to discriminative training of large scale Gaussian mixture HMMs is proposed in [29]. It uses criteria as maximum mutual information (MMI) and minimum phone error (MPE) [24] in an extended Baum–Welch (EBW) estimation framework [8], [21]. The relationships between these estimation criteria and MCE were explored by different authors. In addition, EBW optimization was shown to be very closely related to batch gradient descent in e.g., [26], which in turn is related to the GPD algorithm analyzed in this paper. Indeed, both are gradient descent algorithms which calculate the gradients and perform the updates at different granularities of the training data. The goal of this section is to highlight some of these, at least high level, similarities to better relate the proposed analysis to practical implementations of discriminative training methods in speech recognition.

Regarding the optimization criteria both the MMI and MCE criteria have related functional forms and their relationships have been discussed in e.g., [5]. In addition, [27] proposed a formulation for discriminative training which includes both MMI and MCE as special cases. In [1], it was shown that the difference between MCE and a criterion related to MMI is in the form of the weighting function. While MCE tends to give a Gaussian like weighting to observations which emphasizes observations near the decision boundary, MMI tends to produce a hinge like weighting which favors outliers, i.e., observations having very low discriminative scores. Although the MPE criterion is defined in a different way using a Levenstein distance it aims at minimizing a smoothed estimate of the phone error rate in the same spirit of MCE. Interestingly, a recent study [25] compared a large variety of different discriminative criteria in large scale ASR experiments, and showed only minor differences in performance as long as care is taken in properly handling the corresponding optimization. To summarize, it is expected that different discriminative criteria can share similar properties with the MCE objective that is analyzed in this paper.

In addition to the similarity of other discriminative objective functions to MCE, it is known that extended Baum–Welch optimization is very closely related to batch gradient descent. It is shown in [26] that when the constant D, that is used in EBW re-estimation, is related in a certain way to the step size of batch gradient descent, both algorithms will lead to similar estimates of the means, and variance estimates that differ in a mean-dependent correction factor. Also, in the process of deriving EBW equations, [14] shows that Baum–Welch re-estimation formulas are equivalent to batch gradient descent, for sufficiently large D, to a first-order approximation. In turn, batch estimation is argued in this paper to have the same transient behavior as the GPD algorithm in Section III-D. This leads us to postulate that both GPD and extended Baum–Welch re-estimation may be expected to share similar learning behavior.[4]

The previous qualitative discussion suggests that the MCE/GPD algorithm and the above mentioned discriminative

[4]In the context of batch gradient descent [18], it is possible to use second-order methods like RPROP or Quickprop that potentially lead to faster convergence. It remains an interesting issue to see if the analysis in the paper could be extended to these methods.

training algorithms share conceptual similarities in both the objective function and the optimization procedure. Thus, they are expected to share similarities in their learning behavior. However, more developments are needed to theoretically support these conclusions.

VI. Summary

In this paper, we have performed statistical analysis of a simple discriminative learning algorithm. The algorithm is based on applying the well-known MCE/GPD framework to estimating the means for a simple classification problem consisting of two Gaussian classes with common variance. The relatively simple form of the resulting algorithm allows the use of a framework originally developed for the analysis of adaptive algorithms to derive difference equations for the mean and the decision threshold evolution during learning. These difference equations are applied to study the convergence behavior of the decision threshold, and a proof of the convergence of the decision threshold to its optimal value is given together with an estimate of the number of iterations needed to approach the optimal threshold. In addition, closed-form expressions for the expected values of the smoothed and true error are obtained. Using these expressions, we first show that the expected value of the true error decreases until it saturates at the Bayes error when the threshold reaches its optimal value. We also prove that, under some conditions, the expected smoothed error is an upper bound to the true error and that it approaches it when the ratio of the absolute distance between the class means and the variance tends to infinity. Thus, the expected smoothed error objective function continues to decrease even after the true error converges to its minimum, and the absolute distance between the class means continues to increase without contributing to the decrease of the classification error. This mean drift is then related to the increase of the variance of the classifier. Based on the resemblance of the analyzed algorithm to the MCE/GPD updates of the means of Gaussian mixture HMMs, we experimentally studied MCE/GPD learning of Gaussian mixture HMMs for E-set recognition. We found that the behavior of the learning qualitatively agrees with the pattern suggested by the theoretical analysis. However, some work still needs to be done to formally support this observation. We also qualitatively discussed the relationships between the analyzed MCE/GPD and other discriminative training frameworks, and argued that it is likely that both will share the same learning behavior.

One obvious application of the results obtained in this paper is the use of the normalized inter-class means distance $|\Delta^{(j)}\mu(n)|/\sigma$, or its square, as a penalty term during discriminative optimization to alleviate the mean drift. This additional term can be interpreted as a form of regularization that prevents the increase of the classifier variance. The regularization term is derived in closed form for the simple Gaussian classification problem as noted above. Regularization terms in the same spirit can be motivated for Gaussian mixture or HMMs. Interestingly, this agrees with the use of H-criteria [7] or more recently I-smoothing [24]. In these methods, the discriminative objective function is interpolated with a likelihood based objective function. Although these do not directly agree with the inter-class distance derived in this work, this interpolation can be related to the mean drift observed in this paper. It is easily seen that interpolating with a likelihood function will help in keeping the means close to their original ML estimates, and hence will indirectly reduce the mean drift. From this point of view, these H-criteria can be considered as a form of regularization of the discriminative objective function, to ensure that the model parameters stay close to their ML estimates. In this context, it is worth mentioning that the performance of the MPE criterion [24], a very popular algorithm for training large scale HMMs, is significantly improved using I-smoothing. Also, in the recently proposed maximum margin training [11], it is shown that a penalty term, in the same spirit of regularization, is needed for the success of the training. We plan to explore this line of thought in our future work.

Appendix A

In this appendix, we will prove the result in (7). Taking the expected value of both sides of (5), with the appropriate conditioning, it is straightforward to arrive at

$$\begin{aligned}
&E[\mu_j(n+1)|\mu(n), x(n+1)\in C_j] \\
&\quad = \mu_j(n) \\
&\qquad + \frac{\epsilon}{\sigma^2}E[\phi(d_j(x(n+1),\mu(n)))(x(n+1)-\mu_j(n))] \\
&\quad = \mu_j(n) + \frac{\epsilon}{\sigma^2}E[\phi(a_3v_j+a_1)(\sigma z(n+1)+a_4)] \\
&\quad = \mu_j(n) + \frac{\epsilon}{\sigma}E[\phi(a_3v_j+a_1)z(n+1)] \\
&\qquad + \frac{\epsilon a_4}{\sigma^2}E[\phi(a_3v_j+a_1)] \qquad (30)
\end{aligned}$$

where we have, for convenience, removed the conditioning from the expectations on the right-hand side. The following definitions have been used in (30)

$$v_j = \frac{d_j(x(n+1),\mu(n))-a_1}{a_3} \qquad (31a)$$

$$z(n+1) = \frac{x(n+1)-\mu_j(n)-a_4}{\sigma} \qquad (31b)$$

where for convenience we have defined $a_1 \equiv E[d_j(x(n+1),\mu(n))]$, $a_3^2 \equiv Var[d_j(x(n+1),\mu(n))]$, and $a_4 \equiv E[x(n+1)-\mu_j(n)] = \theta_j - \mu_j(n)$. The values of these and other relevant statistics are given in Appendix B. We now move to the calculation of the two expectations on the right-hand side of (30).

Using Bussgang theorem [22],[5] and noting that both v_j and $z(n+1)$ are zero mean, we can write the first expectation as

$$\begin{aligned}
&E[\phi(a_3v_j+a_1)z(n+1)] \\
&\qquad = E[v_jz(n+1)]E[\phi'(a_3v_j+a_1)]. \qquad (32)
\end{aligned}$$

[5]The theorem states that for two zero mean Gaussian random variables x and y we have $E[yf(x)] = E[xy]E[f'(x)]$.

Evaluating the expectations on the right-hand side of (32), we have for the first expectation

$$\begin{aligned}E[v_j z(n+1)] &= \frac{1}{a_3\sigma}E[(d_j(x(n+1),\mu(n)) - a_1) \\ &\qquad \times (x(n+1) - \mu_j(n) - a_4) \\ &= \frac{1}{a_3\sigma}(E[d_j(x(n+1),\mu(n)) \\ &\qquad \times (x(n+1) - \mu_j(n))] - a_1 a_4) \\ &= \frac{1}{a_3\sigma}\left[\Delta^{(j)}\mu(n) + a_1 a_4 - a_1 a_4\right] \\ &= \frac{\Delta^{(j)}\mu(n)}{|\Delta^{(j)}\mu(n)|} \end{aligned} \tag{33}$$

where the first line follows directly from the definitions in (31a) and (31b), the second line from the definition of the covariance, and the third line from the statistics given in Appendix B. By applying the normal identity [23]

$$\int_{-\infty}^{\infty}\phi'(ax+b)\phi(x)dx = \frac{-ab}{(1+a^2)\sqrt{1+a^2}}\phi\left(\frac{b}{\sqrt{1+a^2}}\right) \tag{34}$$

and noting that v_j has zero mean and unity variance, we evaluate the second expectation as

$$E[\phi'(a_3 v_j + a_1)] = \frac{-a_3 a_1}{(1+a_3^2)\sqrt{1+a_3^2}}\phi\left(\frac{a_1}{\sqrt{1+a_3^2}}\right). \tag{35}$$

Combining (32), (33), and (35), we arrive at the required expectation. The second expectation on the right-hand side of (30) can be calculated using the normal identity [23]

$$\int_{-\infty}^{\infty}\phi(ax+b)\phi(x)dx = \frac{1}{\sqrt{1+a^2}}\phi\left(\frac{b}{\sqrt{1+a^2}}\right) \tag{36}$$

and noting that v_j has zero mean and unit variance as

$$E[\phi(a_3 v_j + a_1)] = \frac{1}{\sqrt{1+a_3^2}}\phi\left(\frac{a_1}{\sqrt{1+a_3^2}}\right). \tag{37}$$

Substituting these expectations into (30), and after some simplification, we arrive at the required result.

Appendix B

This appendix contains some statistics relevant to the derivations in this paper. These can be easily calculated by noting that $d_j(x(n+1),\mu(n))$ is a linear function of $x(n+1)$ and using the well-known properties $E[ax+b] = aE[x]+b$, and $Var[ax+b] = a^2 Var[x]$, and some simple algebra

$$\begin{aligned}&E[d_j(x(n+1),\mu(n))|\mu(n), x(n+1)\in C_j] \\ &\quad = d_j(\theta_j,\mu(n)) \equiv a_1 \end{aligned} \tag{38}$$

$$\begin{aligned}&Var[d_j(x(n+1),\mu(n))|\mu(n), x(n+1)\in C_j] \\ &\quad = \left(\frac{\Delta^{(j)}\mu(n)}{\sigma}\right)^2 \equiv a_3^2. \end{aligned} \tag{39}$$

We note that as the standard deviation is always positive, we define $a_3 \equiv |\Delta^{(j)}\mu(n)|/\sigma$.

$$\begin{aligned}&E[d_j(x(n+1),\mu(n))(x(n+1)-\mu_j(n))|\mu(n), x(n+1)\in C_j] \\ &\qquad = \Delta^{(j)}\mu(n) + (\theta_j - \mu_j(n))d_j(\theta_j,\mu(n)). \end{aligned} \tag{40}$$

The above statistics can be similarly calculated when conditioning on $x(n+1)\in C_{1-j}$, and will be omitted for brevity. We would like to point out that the common variance assumption in Section II leads to a linear misclassification measure $d_j(x(n+1),\mu(n))$, which is also Gaussian, and hence it facilitates the application of the Bussgang theorem, and normal identities as in Appendix A. If this assumption is to be relaxed, the misclassification measure will be quadratic and more approximations will be needed to proceed.

Appendix C

In this appendix, we first derive (26) then we outline how the expected value $E\left[r_j^2(x(n+1),\mu(n))|\mu(n), x(n+1)\in C_j\right]$ is evaluated by a technique similar to Appendix 1.

First, by definition, we have

$$Var[t(n+1)|\mu(n)] = E[t^2(n+1)|\mu(n)] - E^2[t(n+1)|\mu(n)] \tag{41}$$

and using (24) we get

$$\begin{aligned}&E[t(n+1)|\mu(n)] \\ &\quad = t(n) + P_j E[r_j(x(n+1),\mu(n))|\mu(n), x(n+1)\in C_j] \\ &\qquad - P_{1-j}E[r_j(x(n+1),\mu(n))|\mu(n), x(n+1)\in C_{1-j}]. \end{aligned} \tag{42}$$

This is easily seen using (24) and noting that $E[t(n)|\mu(n)] = t(n)$.

Now, expanding the expectation $E[t^2(n+1)|\mu(n)]$ using (24), we can write (43), as shown at the top of the next page, where the second line follows by noting that P_j and P_{1-j} sum to one, and the third line follows from (42). Substituting (43) into the definition of (41) and completing the square we arrive at the required result.

Next, we outline how to calculate the second and third expectations in the last line of (43). We consider only the second expectation and the third similarly follows

$$\begin{aligned}&E\left[r_j^2(x(n+1),\mu(n))|\mu(n), x(n+1)\in C_j\right] \\ &\quad = \left(\frac{\epsilon}{2\sigma^2}\right)^2 (\Delta^{(j)}\mu(n))^2 \\ &\qquad \times E\left[\phi^2(d_j(x(n+1),\mu(n)))|\mu(n), x(n+1)\in C_j\right] \\ &\quad = \left(\frac{\epsilon}{2\sigma^2}\right)^2 (\Delta^{(j)}\mu(n))^2 \left(\frac{1}{\sqrt{2\pi}}\right) \\ &\qquad \times E\left[\phi(\sqrt{2}d_j(x(n+1),\mu(n)))|\mu(n), x(n+1)\in C_j\right] \end{aligned} \tag{44}$$

where the first equality follows from the definition in (24), and the second equality from the observation that $\phi^2(x) = (1/\sqrt{2\pi})\phi(\sqrt{2}x)$. The expectation in the second line

$$
\begin{aligned}
&E[t^2(n+1)|\mu(n)] \\
&\quad = P_j E\Big[t^2(n) + 2t(n) r_j(x(n+1),\mu(n)) + r_j^2(x(n+1),\mu(n))|\mu(n), x(n+1)\in C_j\Big] \\
&\qquad + P_{1-j} E\Big[t^2(n) - 2t(n) r_j(x(n+1),\mu(n)) + r_j^2(x(n+1),\mu(n))|\mu(n), x(n+1)\in C_{1-j}\Big] \\
&\quad = t^2(n) + 2t(n)(P_j E[r_j(x(n+1),\mu(n))|\mu(n), x(n+1)\in C_j] - P_{1-j} E[r_j(x(n+1),\mu(n))|\mu(n), x(n+1)\in C_{1-j}]) \\
&\qquad + P_j E\left[r_j^2(x(n+1),\mu(n))|\mu(n), x(n+1)\in C_j\right] + P_{1-j} E\left[r_j^2(x(n+1),\mu(n))|\mu(n), x(n+1)\in C_{1-j}\right] \\
&\quad = t^2(n) + 2t(n)(E[t(n+1)|\mu(n)] - t(n)) + P_j E\left[r_j^2(x(n+1),\mu(n))|\mu(n), x(n+1)\in C_j\right] \\
&+ P_{1-j} E\left[r_j^2(x(n+1),\mu(n))|\mu(n), x(n+1)\in C_{1-j}\right] \qquad (43)
\end{aligned}
$$

can be evaluated using an exactly similar way to the expectation in (37). This will not be repeated here for brevity.

APPENDIX D

In this appendix, we will prove Theorem 1 of Section III-A. We will prove the theorem for $\Delta t(n) > 0$, and the case of $\Delta t(n) < 0$ can be similarly proved. By subtracting the optimal threshold value t from both sides of (12) and using the definitions in (13)–(15), we can write

$$
\begin{aligned}
\Delta t(n+1) = \Delta t(n) + \frac{0.25\epsilon\alpha(n)}{\sigma} \\
\times \left[\phi\left(\frac{\alpha(n)(\Delta\theta - \Delta t(n))}{\sigma}\right)\right. \\
\left. - \phi\left(\frac{\alpha(n)(\Delta\theta + \Delta t(n))}{\sigma}\right)\right]. \qquad (45)
\end{aligned}
$$

Making use of the inequality[6] $\phi(a-x)-\phi(a+x) \le 2ax/\sqrt{2\pi}$ in (45), and taking $a = \alpha(n)\Delta\theta/\sigma$, and $x = \alpha(n)\Delta t(n)/\sigma$, we get

$$
\begin{aligned}
\Delta t(n+1) &\le \Delta t(n) + \frac{0.5\epsilon\alpha^3(n)\Delta\theta}{\sqrt{2\pi}\sigma^3}\Delta t(n) \\
&= \Delta t(n) - \frac{0.5\epsilon|\alpha^3(n)|\Delta\theta}{\sqrt{2\pi}\sigma^3}\Delta t(n) \\
&\le \Delta t(n) - \frac{0.5\epsilon|\alpha^3(0)|\Delta\theta}{\sqrt{2\pi}\sigma^3}\Delta t(n) \\
&= (1-\gamma)\Delta t(n) \qquad (46)
\end{aligned}
$$

where the second line follows because $\alpha^3(n)$ is negative, the third line is obtained by noting that $|\alpha^3(n)|$ is an increasing function of n, and the fourth line results by defining $\gamma \equiv (0.5\epsilon|\alpha^3(0)|\Delta\theta/\sqrt{2\pi}\sigma^3)$.

Now, applying (46)n times starting from iteration zero, we finally obtain

$$\Delta t(n) \le (1-\gamma)^n \Delta t(0) \qquad (47)$$

where $\Delta t(0)$ is the initial distance from the optimal threshold. For $\gamma < 1$ and $\Delta t(0)$ is finite, we take the limit of the right-hand side as $n \to \infty$ to yield $\Delta t(n) \le 0$. Because $\Delta t(n)$ is positive by assumption, this implies that $\Delta t(n) = 0$, which is the required convergence result.

[6]This can be proved by noting that $\phi(a-x) \le 1/\sqrt{2\pi}$, and that e^{-2ax} is a convex function and hence lies above its first-order Taylor's expansion.

Further, if γ is sufficiently small such that $(1-\gamma)^n \approx 1-n\gamma$, then $n \approx 1/\gamma$ can be considered as an estimate of the number of iterations needed to reach steady state. This completes the proof.

ACKNOWLEDGMENT

The authors would like to thank Prof. C.-H. Lee of Georgia Tech for his insightful comments on an initial version of this paper, and the anonymous reviewers for carefully reviewing the paper and providing very helpful suggestions.

REFERENCES

[1] M. Afify and O. Siohan, "A discriminative training criterion and an associated EM learning algorithm," in *Proc. ICASSP'02*, Orlando, FL.
[2] M. Afify, X. W. Li, and H. Jiang, "Statistical performance analysis of MCE/GPD learning in Gaussian classifiers and Hidden Markov models," in *Proc. ICASSP'05*, Philadelphia, PA.
[3] N. Bershad, J. Shynk, and P. Feintuch, "Statistical analysis of single-layer back-propagation algorithm: Part I- mean weight behavior," *IEEE Trans. Signal Process.*, vol. 41, pp. 573–582, Feb. 1993.
[4] A. Biem, "Discriminative Feature Extraction Applied to Speech Recognition," Ph.D. dissertation, Univ. Paris, Paris, France, 1997.
[5] W. Chou, "Discriminant-function-based minimum recognition error rate pattern-recognition approach to speech recognition," *Proce. IEEE*, vol. 88, no. 8, pp. 1201–1223, Aug. 2000.
[6] R. Duda, P. Hart, and D. Stork, *Pattern Classif.*, S. Edition, Ed. New York: Wiley-Interscience, 2000.
[7] P. S. Gopalakrishnan, D. Kanevsky, A. Nadas, D. Nahamoo, and M. A. Picheny, "Decoder selection based on cross entropies," in *Proc. ICASSP'88*, New York.
[8] P. S. Gopalakrishnan, D. Kanevsky, A. Nadas, and D. Nahamoo, "An inequality for rational functions with applications to some statistical estimation problems," *IEEE Trans. Inf. Theory*, vol. 37, no. 1, pp. 107–113, Jan. 1991.
[9] H. S. M. Beigi, S. H. Maes, and J. S. Sorensen, "A distance measure between collections of distributions and its application to speaker recognition," in *Proc. ICASSP'98*, Seattle, WA, Apr. 1998.
[10] C.-S. Huang, H.-C. Wang, and C.-H. Lee, "A study on model-based error rate estimation for automatic speech recognition," *IEEE Trans. Speech Audio Process.*, vol. 11, no. 6, pp. 581–589, Nov. 2003.
[11] H. Jiang, X. Li, and C.-J. Liu, "Large margin Hidden Markov models for speech recognition," *IEEE Trans. Speech Audio Process.*, vol. 14, no. 5, pp. 1584–1595, Sep. 1996.
[12] B.-H. Juang and S. Katagiri, "Discriminative learning for minimum error classification," *IEEE Trans. Signal Process.*, vol. 40, no. 12, pp. 3043–3054, Dec. 1992.
[13] B.-H. Juang, W. Chou, and C. H. Lee, "Minimum classification error rate methods for speech recognition," *IEEE Trans. Speech Audio Process.*, vol. 5, no. 3, pp. 257–265, May 1997.
[14] D. Kanevsky, "Extended Baum transformations for general functions," in *Proc. ICASSP'04*, Montreal, QC, Canada.
[15] S. Katagiri, B. H. Juang, and C. H. Lee, "Pattern recognition using a family of design algorithms based upon the generalized probabilistic descent method," *Proc. IEEE*, vol. 86, no. 11, pp. 2345–2373, Nov. 1998.

[16] S. Katagiri, C. H. Lee, and B. H. Juang, "New discriminative training algorithm based on the generalized probabilistic descent method," in *Proc. 1991 IEEE Workshop on Neural Networks for Signal Processing*, pp. 299–307.
[17] E. Mcdermott, "Discriminative Training for Speech Recognition," Ph.D. dissertation, Waseda University, Tokyo, Japan, 1997.
[18] J. L. Roux and E. Mcdermott, "Optimization methods for discriminative training," in *Proc. EUROSPEECH'05*, Lisbon, Portugal, Sep. 2005, pp. 2941–2944.
[19] E. McDermott and S. Katagiri, "A new formalization of minimum classification error using a parzen estimate of classification chance," in *Proc. ICASSP'03*, Hong Kong, Apr. 2003.
[20] A. Nadas, "A decision theoretic formulation of a training problem in speech recognition and a comparison of training by unconditional versus conditional maximum likelihood," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 31, no. 4, pp. 814–817, Aug. 1983.
[21] Y. Normandin, "Hidden Markov Models, Maximum Mutual Information Estimation and the Speech Recognition Problem," Ph.D. dissertation, McGill University, Montreal, QC, Canada, 1991.
[22] A. Papoulis and S. U. Pillai, *Probability, Random Variables, and Stochastic Processes*. New York: McGraw-Hill, 2002.
[23] J. K. Patel and C. B. Read, *Handbook of the Normal Distribution*. New York: Marcel Decker, 1996.
[24] D. Povey and P. C. Woodland, "Minimum phone error and I-smoothing for improved discriminative training," in *Proc. ICASSP'02*, Orlando, FL.
[25] D. Povey and B. Kingsbury, "Evaluation of proposed modifications to MPE for large scale discriminative training," in *Proc. ICASSP'07*, Honolulu, HI, Apr. 2007.
[26] R. Schlueter, W. Macherey, S. Kanthak, H. Ney, and L. Welling, "Comparison of optimization methods for discriminative training criteria," in *Proc. EUROSPEECH'97*, Rhodes, Greece, 1997.
[27] R. Schlueter and W. Macherey, "Comparison of discriminative training criteria," in *Proc. ICASSP'98*, Seattle, WA, May 1998, pp. 817–820.
[28] R. Schlueter and H. Ney, "Model-based MCE bound to the true Bayes' error," *IEEE Signal Process. Lett.*, vol. 8, no. 5, pp. 131–133, May 2001.
[29] P. C. Woodland and D. Povey, "Large scale discriminative training of Hidden Markov models for speech recognition," *Comput. Speech Lang.*, vol. 16, pp. 25–48, 2002.

Mohamed Afify was born in Cairo, Egypt, in 1964. He received the B.Sc. degree (with distinction), M.Sc., and Ph.D. degrees from the Department of Electronics and Communications, Cairo Univeristy in 1987, 1992, and 1995, respectively.

From 1989 to 1996, he was a Research Associate at the National Telecommunication Institute, Cairo. From 1996 to 1998, he was a Postdoctoral Research Fellow with the Speech Recognition Group, INRIA Lorraine, Nancy, France. From 1998 to 2000, he was an Assistant Professor with the Department of Electrical Engineering, Cairo Univerdsity, Fayoum Branch. From 2000 to 2002, he joined the Dialogue Systems Research Department at Bell Laboratories as a consultant. Since 2002, he has been an Associate Professor with the Faculty of Information and Computer, Cairo University. From March 2004 to June 2005 he was with BBN Technologies, Cambridge, MA, working on large-vocabulary speech recognition. In June 2005, he joined the IBM T. J. Watson Research Center, Yorktown Heights, NY. His research interests are in statistical modeling, pattern recognition, and digital signal processing with a particular emphasis on their application to speech and language processing.

Xinwei Li received the B.S. degree in electronics from Beijing University, Beijing, China, and the M.S. degree in computer science from York University, Toronto, ON, Canada.

He is a Speech Scientist with Nuance, Inc., Burlington, MA. His major research interest focuses on automatic speech recognition, especially discriminative training.

Hui Jiang (M'00) received the B.Eng. and M.Eng. degrees from the University of Science and Technology of China (USTC), Hefei, and the Ph.D. degree from the University of Tokyo, Tokyo, Japan, in September 1998, all in electrical engineering.

From October 1998 to April 1999, he worked as a Researcher at the University of Tokyo. From April 1999 to June 2000, he was with Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON, Canada, as a Postdoctoral Fellow. From 2000 to 2002, he worked in Dialogue Systems Research, Multimedia Communication Research Lab, Bell Labs, Lucent Technologies, Inc., Murray Hill, NJ. He joined the Department of Computer Science and Engineering, York University, Toronto, ON, Canada, as an Assistant Professor in fall 2002 and was promoted to Associate Professor in 2007. His current research interests include speech and audio processing, machine learning, statistical data modeling, and bioinformatics, especially discriminative training, robustness, noise reduction, utterance verification, and confidence measures.